IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Steven D. Clynes, et al. **Art Unit: 2685**

Serial No.: 09/862,523 **Examiner: Aung Soe Moe**

Filed: 05/22/01 **Docket: TI-32423**

For: ON-CHIP 2D ADJUSTABLE DEFECTIVE PIXEL FILTERING FOR CMOS IMAGERS

APPELLANTS' BRIEF UNDER 37 C.F.R. §1.192

Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Commissioner:

The following Appeal Brief is respectfully submitted in connection with the above-identified application in response to the Final Rejection mailed September 9, 2005. Please charge all required fees, including any extension of time fees, to the deposit account of Texas Instruments Incorporated, Deposit Account No. 20-0668.

REAL PARTY IN INTEREST

The real party in interest is Texas Instruments Incorporated, to whom this application is assigned.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to the Applicant's legal representative.

STATUS OF THE CLAIMS

Claims 1, 3-13, 17, 18 and 20 are the subject of this appeal. Claims 1, 3-13, 17, 18 and 20 are rejected. This application was filed on May 22, 2001.

STATUS OF THE AMENDMENTS

The Appellants filed an amendment under 37 C.F.R. § 1.116 on December 6, 2005 in response to the Office Action dated September 9, 2005, with no amendments to the claims. The Appellants filed an amendment under 37 C.F.R. § 1.116 on August 5, 2005 in response to the Office Action dated May 9, 2005. The Appellants filed an amendment under 37 C.F.R. § 1.115 on December 13, 2004 in response to the Office Action dated August 12, 2004.

SUMMARY OF CLAIMED SUBJECT MATTER

The invention provides on-chip real time elimination of the effect of white and dark pixels, and thus increases the imager yield and decrease CMOS Imager cost. The invention comprises scanning each of a plurality of pixels within a block, and designating a pixel as a process pixel. The process pixel is then compared to adjacent pixels to see if the process pixel value deviates significantly from an adjacent pixel value, to determine if error correction is needed. If error correction is needed then the method provides for error correction. Accordingly, the invention is an On-Chip 2D Adjustable Defective Pixel Filtering (2DOCADPF) method of real-time filtering out defective pixels (or other similar spot-noise).

On Chip pixel defect correction may be better understood by the examination of an adjustable defective pixel correction algorithm. Figure 3 is a block-flow diagram of a pixel correction algorithm 300. Pixel correction algorithm 300 begins with a capture pixel act 310 in which a pixel value is transferred to a register. It should be understood that, although the word capture is used to describe the capture pixel act, the term "capture" does not necessarily imply the loading of a pixel value into a memory, as may be implied in some prior art. After a pixel value is transferred to a register in the capture pixel act 310, the pixel correction algorithm 300 proceeds to a register full query 320. The register full query 320 determines whether the registers that store adjacent pixel values (which are needed for determining an average pixel value to correct a process

pixel) do in fact have pixel values associated with them. Accordingly, if each such register has a pixel value associated with it, then the pixel correction algorithm 300 proceeds along the y (or "yes") path to a compare query 330. If, however, additional registers need pixel values for proper process pixel processing, then the pixel correction algorithm 300 proceeds along the n ("no") path and returns to the capture pixel act 310 where another pixel value is transferred to a register.

Accordingly, the pixel correction algorithm 300 loads the registers that are needed for proper process pixel processing and then proceeds to compare a process pixel value to the pixel values of the adjacent pixels in the compare query 330. If in the compare query 330 it is determined that the process pixel does not lie outside of a threshold range beyond a highest adjacent pixel value, and that the process pixel value does not lie outside of a threshold range beyond a lowest adjacent pixel value, then the pixel correction algorithm 300 returns to the capture pixel act 310 and another pixel value is transferred to a register. If, however, the compare query 330 determines that the process pixel value lies outside of a threshold beyond either a highest adjacent pixel value or a lowest adjacent pixel value, then the pixel correction algorithm 300 proceeds along the y path to a reset act 340.

In the reset act 340 the process pixel is replaced by a new pixel value. Preferably, the new pixel value is the average pixel value of the adjacent pixels. However, it should be understood that the new process pixel value may be determined in other ways as discussed below. It should be noted that the horizontal nature of the flow of Figure 3 is shown to emphasize that the preferred processes of Figure 3 are occurring simultaneously (or, in parallel), preferably as a result of on-chip processing.

However, it should be understood that the pixel correction algorithm 300 may also be accomplished as software.

Figure 4 provides a logic-flow for a pixel defect correction algorithm 400. The pixel defect correction algorithm 400 begins with an exposed image act 410 in which an array is exposed to light so that each pixel in the array generates an electric impulse (or, electric signal) indicative of the light intensity and wave length that reaches the pixel. These pixel characteristics are thus embodied as an electronic pixel value. However, it should be understood that the pixel value may indicate color, brightness, darkness, contrast, and/or a variety of other pixel qualities.

Next, in a capture pixel act 420, the pixel value is transferred to a register preferably co-located on the same chip as the array. The capture pixel act 420 preferably begins by capturing the top left most pixel in the array. The register full query 430 determines whether the registers that store adjacent pixel values (which are needed for determining an average pixel value to correct a process pixel) do in fact have pixel values associated with them. Accordingly, if each such register has a pixel value associated with it, then the pixel defect correction algorithm 400 proceeds along the y (or "yes") path to a compare query 450. If, however, additional registers need pixel values for proper process pixel processing, then the pixel correction algorithm 300 proceeds along the n ("no") path advances to a move act 440.

The move act 440 increments to the next pixel of the array, preferably incrementing horizontally from left to right in rows, so that when the last pixel in a row is captured, the move act 440 returns to the far left of the array and increments one row

down from the previously captured row. Then, after the move act 440, the pixel defect correction algorithm 400 returns to the capture pixel act 420.

Accordingly, the pixel defect correction algorithm 400 loads the registers that are needed for proper process pixel processing and then proceeds to compare a process pixel value to the adjacent pixel values in a compare query 450. If in the compare query 450 it is determined that the process pixel does not lie outside of a threshold range beyond a highest adjacent pixel value, and that the process pixel value does not lie outside of a threshold range beyond a lowest adjacent pixel value, then the pixel defect correction algorithm 400 advances to the move act 440. If, however, the compare query 450 determines that the process pixel value lies outside of a threshold beyond either a highest adjacent pixel value or a lowest adjacent pixel value, then the pixel defect correction algorithm 400 proceeds along the y path to an average adjacent pixels act 460.

In the average adjacent pixel values act 460 the pixel values for the pixels adjacent to the process pixel (or their substitutions, as discussed below) are average. However, it should be understood that in the average adjacent pixel values act other forms of pixel value substitutions for the process pixel may be selected by a user. Then, the pixel defect correction algorithm 400 proceeds to a reset process pixel value act 470 in which the process pixel value that was captured is replaced (or reset) to the pixel value generated in the average adjacent pixel values act 460. Then, the pixel defect correction algorithm 400 advances to the move act 440.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Rejection under 35 U.S.C. § 102 (a) as being anticipated by European Patent Application EP 1,045,578 A2.

ARGUMENT

Rejection under 35 U.S.C. § 102 (a) as being anticipated by European Patent Application EP 1,045,578 A2

Claims 1, 3-13, 17-18, and 20

Claim 1 includes "... detecting a lowest pixel value among the adjacent pixels." Claim 17 includes "...detecting the lowest pixel value among the adjacent pixels." European Patent Application EP 1,045,578 A2 does not show, teach, or suggest the above recited limitations of claims 1 and 17. European Patent Application EP 1,045,578 A2 does not teach or suggest detecting the lowest pixel value among the adjacent pixels. European Patent Application EP 1,045,578 A2 compares a primary pixel to adjacent pixels, but does not compare adjacent pixels to each other to

determine a lowest pixel value among the adjacent pixels.

CONCLUSION

For the foregoing reasons, Appellants respectfully submit that the Examiner's final rejection of Claims 1, 3-13, 17, 18 and 20 is improper, and it is respectfully requested that the Board of Patent Appeals and Interferences so find and reverse the Examiner's rejection.

Please charge any fees necessary in connection with the filing of this paper, including any necessary extension of time fees, to Deposit Account No. 20-0668 of Texas Instruments Incorporated.

Respectfully submitted,

/Alan K. Stewart/
Alan K. Stewart
Attorney for Appellants
Registration No. 35,373

Texas Instruments, Incorporated
P. O. Box 655474 - M/S 3999
Patent Department
Dallas, Texas 75265
972/917-5466

CLAIMS APPENDIX

1. A method of pixel filtering for CMOS imagers, comprising:

scanning each of a plurality of pixels within a block;

designating a pixel as a process pixel, the process pixel having adjacent pixels, the process pixel having a process pixel value, each of the adjacent pixels having an adjacent pixel value;

comparing the process pixel value to at least one adjacent pixel value; and

detecting a lowest pixel value among the adjacent pixels.

3. The method of claim 1 wherein comparing compares the process pixel value to a lowest pixel value.

4. The method of claim 3 further comprising resetting the process pixel to a new pixel value.

5. The method of claim 4 wherein the new pixel value is the average pixel value of the adjacent pixel values.

6. The method of claim 1 further comprising detecting a highest pixel value among the adjacent pixels.

7. The method of claim 6 wherein comparing compares the process pixel value to a highest pixel value.

8. The method of claim 7 further comprising resetting the process pixel value when the process pixel value is a predetermined value lower than the lowest pixel value.

9. The method of claim 3 further comprising resetting the process pixel value when the process pixel value is a predetermined value greater than a highest pixel value.

10. The method of claim 1 further comprising exposing an array to a light source so as to cast an image on the array, the array having at least one block.

11. The method of claim 10 wherein the array is generally grid-shaped.

12. The method of claim 1 wherein the block is generally grid-shaped.

13. The method of claim 12 wherein the block has nine pixels.

17. A method of on-chip pixel filtering for CMOS imagers, comprising:

scanning each of a plurality of pixels within a block for a pixel value;

loading a pixel value into a register;

using filter logic to designate a pixel as a process pixel, the process pixel having adjacent pixels, the process pixel having a process pixel value, each of the adjacent pixels having an adjacent pixel value; and

using filter logic to compare the process pixel value to at least one adjacent pixel value,

wherein the filter logic compares the process pixel value to a lowest pixel value, further comprising:

detecting the lowest pixel value among the adjacent pixels; and

resetting the process pixel value to a new process pixel value when the process pixel value is a predetermined value lower than the lowest pixel value.

18. The method of claim 17, wherein the filter logic compares the process pixel value to a highest pixel value, further comprising:

detecting the highest pixel value among the adjacent pixels; and

resetting the process pixel value to a new process pixel value when the process pixel value is a predetermined value higher than the highest pixel value.

20. The method of claim 17 wherein the new process pixel value is the average pixel value of the adjacent pixel values.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.